

CONTENTS

Basics	2
Webhook Events	2
Event Payloads	2
Configuration	2
Events.....	3
New incoming message & Include Own Message.....	3
Case Closed.....	5
Changes in personID	6
Troubleshooting.....	7
Webhook disabled.....	7



BASICS

Webhooks are custom HTTP call-backs that can be used to receive data. Webhooks are triggered by an event. In Tracebuzz we have the functionality to connect webhooks. When an event has been triggered, it will send a payload to the configured webhook.

WEBHOOK EVENTS

The following events are supported.

Event type	Trigger
New incoming message	New customer message has been received in Tracebuzz
Include own message	New agent message has been received in Tracebuzz *Can only be used with Event New incoming message.
Case closed	Agent has closed a case
Changes in personID	Person has been updated [for example when the user is authenticating from anonymously user to registered user]

EVENT PAYLOADS

When an event is triggered, it will send a JSON Payload to the webhook(s) configured to this event.

CONFIGURATION

Users with "configuration access" enabled can configure webhooks.

Each profile has its own webhook(s).

To modify or create a webhook go to Profile -> Webhooks. The webhook will be tested when you save the configuration and saved successfully when a response status 200 OK is received.

Name	A name to identify your webhook
Type webhook	The triggers that will send the payloads to the url. You can select multiple triggers for one endpoint.
URL	The endpoint the payload will be sent to.
Email [optional]	Notification will be sent to this email if webhook fails.
Custom header name & Custom header content [optional]	Add Custom authorization to the webhook.

EVENTS

NEW INCOMING MESSAGE & INCLUDE OWN MESSAGE

This is triggered when a new message has been received in Tracebuzz.

A new message from an agents will only be triggered if "Include own message" is also set else only new customer messages will be triggered.

Payload description

Key	Description	Type
event	Describe the event name	string
searchId	The identifier of the search profile	integer
uniqueId	The identifier of the message itself	integer
resultId	The identifier of the message itself, same as the UniqueId	integer
personId	The person this conversation is about	integer
caseId	The case identifier inside the conversation	integer
handlerId	Identifier of the current case	integer
sourceId	The identifier of the channel that is been used	integer
connectionId	The identifier of the current channel that is configured to the selected account	integer
typeId	The identifier of the message itself. Available values: 1 = page 2 = post //including @mentions 3 = comment 4 = private message 5 = ad 6 = system message	integer
own	Determines if the origin of the message is from the customer or the agent. Available values: 0 = customer 1 = agent	integer
timestamp	Unix timestamp to determine when the event was triggered	integer

New Incoming Message Payload example:

```
{
  "event": "newResult",
  "searchId": 11111,
  "uniqueId": 11111,
  "resultId": 11111,
  "personId": 11111,
  "caseId": 11111,
  "handlerId": 0,
  "sourceId": 14,
  "connectionId": 111,
  "typeId": 4,
  "own": 0,
  "timestamp": 1663831826
}
```

Include own message Payload example:

```
{
  "event": "newResult",
  "searchId": 11111,
  "uniqueId": 11111,
  "resultId": 11111,
  "personId": 11111,
  "caseId": 11111,
  "handlerId": 0,
  "sourceId": 14,
  "connectionId": 111,
  "typeId": 4,
  "own": 1,
  "timestamp": 1663831826
}
```

CASE CLOSED

The event “Case closed” will be triggered when the agent closes the case in the conversation with the customer.

Payload description

Key	Description	Type
event	Describe the event name.	string
searchId	The identifier of the search profile.	integer
caseId	The case identifier inside the conversation.	integer
externalCaseId	The case identifier inside the conversation. This could be used for external systems. When no value available = null.	null, integer
reasonId	The identifier of the reason why the person started this case [conversation]. When no value available = null.	integer
personId	The identifier of the person this conversation is about.	integer
externalPersonId	The identifier of the external person. This could be used for external systems. When no value available = false.	boolean, integer
agentId	Identifier of the current case.	integer
resultId	The identifier of the message that is been used.	integer
connectionId	The identifier of the current channel that is configured to the selected account.	integer
timestamp	Unix timestamp to determine when the event was triggered.	integer

Case Closed Payload example:

```
{
  "event": "caseClosed",
  "searchId": 11111,
  "caseId": 11111111,
  "externalCaseId": null,
  "reasonId": 111111,
  "personId": 11111111,
  "externalPersonId": false,
  "agentId": 1,
  "resultId": 1111111,
  "connectionId": 111,
  "timestamp": 1663845416
}
```

CHANGES IN PERSONID

Key	Description	Type
event	Describe the event name	string
oldPersonId	The identifier of the personId it was before	integer
newPersonId	The identifier of the personId it is now	integer
timestamp	Unix timestamp to determine when the event was triggered	integer

Changes in personID Payload example:

```
{
  "event": "profileMerged",
  "oldPersonId": 1,
  "newPersonId": 1,
  "timestamp": 1663845416
}
```

TROUBLESHOOTING

WEBHOOK DISABLED

When we do not receive a **200 OK response status** from the configured webhook endpoint. We will attempt to send the payload again automatically in the following timetable.

Attempt	Retry after failure
1	5 minutes
2	7 minutes
3	21 minutes
4	59 minutes
5	133 minutes
6	255 minutes
7	437 minutes
8	691 minutes
9	1029 minutes
10	1758 minutes

After 10 attempts the webhook will be disabled. You might need to change the webhook configuration and enable it again by saving the webhook. If a 200 OK response status is received the webhook will be active again.